Just How Did I Get All The Information and Process it For Last Night's Email. Here's your Answer.

From: Joshua Powell <Josh@thepowellhousepress.com>
Date: Monday, February 16, 2026 at 9:40 AM
To: Joshua Powell; and others (100+ recipients)

Good Morning,

I'm getting a lot of questions coming my way. The most common question I getting is: *how did you learn everything about the board's activities? Surely there's too much information to review. Did you use AI?*

No. I didn't, for several reasons. And I am sharing my process so that people can understand what I did, question it, get real answers and provide transparency.

First, reliability. AI-generated research still produces errors, hallucinated citations, and fabricated details. Think RFK Jr.'s MAHA Report. It's not yet a dependable tool for investigative work that demands precision. Second, AI tools cannot dig into a website's file structure, navigate paywalls, bypass password protections, or access the kinds of archived municipal documents that local governments often store in less-than-intuitive ways. What I did use was my computer, some basic programming skills, and a methodical process.

I use Mac products, but the approach is universal across platforms. Think of Macs and PCs as people from the same country who speak slightly different dialects. The underlying logic is the same.

Part One: Getting the Files

The Town of Chatham's website contains hundreds of documents. Some are individual PDFs (PDF stands for Portable Document Format, a file type designed to preserve a document's layout regardless of what device you open it on), each covering a single meeting. For older records, the town has merged multiple meetings into single files organized by year. Opening each document individually and saving it by hand would have taken an enormous amount of time.

There are faster ways to do this, all perfectly legal, no hacking involved.

On a Mac, if you press Command + Spacebar, type "Terminal," and hit Enter, you open a text-based interface where you can write and run code directly. I've been comfortable in this environment for decades. Early in my career, I coded websites to supplement my income. In school, I learned programs like SPSS and SAS for working with large datasets. So none of this was new to me. Am I the best coder out there? No way - no how. I can't launch a bomb from a silo or rob bank accounts - but online public information, well...yeah that's in wheelhouse.

I wrote a simple script in Python, a widely used programming language, to automatically download all the available Town Board and ZBA meeting documents from the website to my computer. I'll

include a portion of the code below. It may look like gibberish, but to the computer, it's a set of clear instructions. I also used a free online code checker to catch my mistakes along the way. Within about 15 minutes, I had copies of meeting records going back years.

Part Two: Searching the Records

Next, I wrote another Python script to search every downloaded PDF for specific keywords and phrases (this was more involved but still accurate), including names like Bashford and Williams, among others. When you account for variations in spelling, formatting, and word combinations, the total number of search permutations ran into the thousands.

I then tested my results for accuracy, cross-referencing flagged documents against the originals, verifying that hits were genuine, and confirming that the process produced consistent results when rerun.

Every document that flagged positive for relevant keywords, I read personally. That's when I began noticing references to "Town and County" appearing in the town meeting records. From there, I went back and read all available meeting minutes through 2024, looking specifically for any disclosures on Donald Collins's part.

I used the same methodology to research the pool fence issue involving the ZBA. Another Python scripts downloaded and searched ALL the ZBA meeting documents from 2010, using a different set of keywords tailored to that matter. The same process of automated retrieval, keyword searching, manual verification, and careful reading of every flagged document applied. The tools didn't change. Only the search terms did. I want to note that there could be a variance, but if there is the processed used to get it was very different from the normal way the ZBA works.

And that is how I did it.

Example of some coding I used:

```
#!/usr/bin/env python3
# scrape chatham ZBA pdfs and search for keywords
# pip3 install pymupdf requests beautifulsoup4
import os, re, sys, time
import requests
from bs4 import BeautifulSoup
try:
    import fitz
except ImportError:
    print("need pymupdf - run pip3 install pymupdf")
    sys.exit(1)
DOWNLOAD_DIR = os.path.expanduser("~/chatham_zba_pdfs")
SCRAPE_URLS = [
    "https://www.chathamnewyork.us/government/zoning_board_of_app..."
]
KEYWORDS = [
```

CONFIDENTIALITY NOTICE

This email and any attachments are intended solely for the use of the individual or entity to whom they are addressed. This communication may contain confidential and/or privileged information. If you are not the intended recipient, you are hereby notified that any disclosure, copying, distribution, or taking of any action in reliance on the contents of this information is strictly prohibited. If you have received this email in error, please immediately notify the sender and delete this email from your system.

© 2025 Josh Powell and the Powell House Press. All rights reserved. No part of this publication may be reproduced, distributed, or transmitted without prior written permission.